

**SPEECH RECOGNITION HANDLING WITH SYNTHESIZED
MANUAL INPUT EVENTS**

Technical Field

The present invention is generally directed to speech
5 recognition and, more specifically, to synthesizing manual input events within
a speech recognition driven system.

Background of the Invention

As is well known to one of ordinary skill in the art, speech
10 recognition is a field in computer science that deals with designing computer
systems that can recognize spoken words. A number of speech recognition
systems are currently available (e.g., products are offered by IBM, Dragon
Systems and Philips). Traditionally, speech recognition systems have only
been used in a few specialized situations due to their cost and limited
15 functionality. For example, such systems have been implemented when a user
was unable to use a keyboard to enter data because the user's hands were
disabled. Instead of typing commands, the user spoke into a microphone.
However, as the cost of these systems has continued to decrease and the
performance of these systems has continued to increase, speech recognition
20 systems are being used in a wider variety of applications (as an alternative to
or in combination with keyboards or other user interfaces). For example,
speech actuated control systems have been implemented in a motor vehicle to
control various automotive accessories within the motor vehicle.

A typical speech recognition system, that is implemented in a
25 motor vehicle, includes voice processing circuitry and memory for storing
data representing command words (that are employed to control various
vehicle accessories). In a typical system, a microprocessor is utilized to
compare the user provided data (e.g., voice input) to stored speech models to
determine if a word match has occurred and provide a corresponding control
30 output signal in such an event. The microprocessor has also normally

controlled a plurality of motor vehicle accessories, e.g., a cellular telephone and a radio. Such systems have advantageously allowed a driver of the motor vehicle to maintain vigilance while driving the vehicle.

In a typical computer system, various software applications may receive notification about keyboard events, such as, a key press, a key hold and/or a key release, via a message provided by an operating system. In response to these messages, an appropriate task is usually initiated by one or more of the applications. Today, many voice recognition applications include separate code, which allows the application to receive both voice and manual input (for example, the manual input may be provided by a switch, a push-button or a rotary dial). This has allowed a user of the speech recognition system to also manually (as opposed to verbally) provide input to the system. However, such systems have typically required increased development time to write and prove-in their voice recognition applications as the applications have to perform a sequence of additional calls to achieve full application level functionality.

What is needed is a speech recognition system that includes a voice recognition application that does not require additional calls to handle a voice input.

Summary of the Invention

According to the present invention, a voice responsive system controls a device in response to both a manual input and a voice input. The system includes a processor, a memory subsystem coupled to the processor and processor executable code. The processor executable code includes an operating system and a voice recognition application for causing the processor to perform a number of steps. Initially, a voice message is provided from the operating system to the voice recognition application. Next, an appropriate simulated manual input event is provided from the voice recognition application to the operating system in response to the received voice message. Then, an appropriate manual input event message is provided from the operating system

to the voice recognition application in response to the simulated manual input
sevent. The voice recognition application then initiates control of the device
responsive to the manual input event message.

These and other features, advantages and objects of the present
5 invention will be further understood and appreciated by those skilled in the art
by reference to the following specification, claims and appended drawings.

Brief Description of the Drawings

The present invention will now be described, by way of
10 example, with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram of an exemplary speech recognition
system implemented in a motor vehicle;

Fig. 2 is a block diagram illustrating the flow of information
from a hardware device to an operating system and a voice recognition
15 application;

Fig. 3 is a block diagram depicting the flow of information
from a microphone to an operating system and a voice recognition application;

Fig. 4 is a flowchart depicting a transfer routine for
transferring information between an operating system and a voice recognition
20 application; and

Fig. 5A depicts an exemplary block diagram illustrating the
execution of calls to perform a function, according to an embodiment of the
present invention; and

Fig. 5B depicts an exemplary block diagram illustrating the
25 execution of calls to perform the function of Fig. 5A, according to the prior
art.

Description of the Preferred Embodiments

The present invention is directed to a speech recognition system
30 that synthesizes a manual input event within a voice recognition application.
When the application receives a voice message (e.g., a voice command), the

application uses an operating system call function to simulate an appropriate manual input event, e.g., a key press, a key hold or a key release, and thereby provide a simulated manual input event to the operating system. In response to receiving the simulated manual input event (via the system call function), the operating system provides an appropriate manual input event message to the application. Upon receiving the manual input event message, the application implements a task that corresponds to the manual input event message, such as, changing the station on a radio. In this manner, the application uses the same code to initiate implementation of a task associated with a manual input and a voice input that corresponds to the manual input.

Fig. 1 depicts a block diagram of an exemplary speech recognition system 100 (implemented within a motor vehicle) that handles an exemplary manual input 101 and a voice input with the same code, according to an embodiment of the present invention. The system 100 includes a processor 102 coupled to a motor vehicle accessory (e.g., a radio) 124 and a display 120. The processor 102 controls the motor vehicle accessory 124 as dictated by a voice input or the manual input 101, supplied by a user of the system 100. The processor 102 also supplies various information to the display 120, to allow a user of the motor vehicle to better utilize the system 100. In this context, the term processor may include a general purpose processor, a microcontroller (i.e., an execution unit with memory, etc., integrated within a single integrated circuit) or a digital signal processor. The processor 102 is also coupled to a memory subsystem 104. The memory subsystem 104 includes an application appropriate amount of main memory (volatile and non-volatile).

An audio input device 118 (e.g., a microphone) is coupled to a filter/amplifier module 116. The filter/amplifier module 116 filters and amplifies the voice input provided by the user through the audio input device 118. The filter/amplifier module 116 is also coupled to an analog-to-digital (A/D) converter 114. The A/D converter 114 digitizes the voice input from

the user and supplies the digitized voice to the processor 102 (which may cause the voice input to be compared to system recognized commands).

The processor 102 may execute various routines in determining whether a voice or manual input corresponds to a system recognized message (i.e., command). The processor 102 may also cause an appropriate voice output to be provided to the user (ultimately through an audio output device 112). When implemented, the synthesized voice output is provided by the processor 102 to a digital-to-analog (D/A) converter 108, which is coupled to a filter/amplifier section 110, which amplifies and filters the analog voice output. The amplified and filtered voice output is then provided to the audio output device 112 (e.g., a speaker). While only one manual input 101 and one motor vehicle accessory 124 are shown, it is contemplated that any number of manual inputs and accessories, typically provided in a motor vehicle (e.g., a cellular telephone or a radio and their corresponding manual inputs), can be implemented.

Fig. 2 depicts a block diagram illustrating the transfer of information between a manual input device (e.g., a keyboard, a switch, a push-button or a rotary dial, etc.) 202, an operating system (e.g., Windows CE) 204 and a voice recognition application 206. As depicted in Fig. 2, when the manual input device 202 is actuated, a manual input event 201 is provided to the operating system 204. In response to the manual input event 201, the operating system 204 provides a manual input event message 203 to the application 206. The application 206 then initiates performance of a task that corresponds to the manual input event message 203.

Fig. 3 depicts a microphone 302 providing a voice input to the operating system 204, which is in communication with the application 206. The microphone 302 receives a voice input from a user and provides a voice event 301 to the operating system 204. In response to the voice event 301, the operating system 204 provides a voice message 303 to the application 206. In response to the voice message 303, the application 206 simulates a manual input event and provides a simulated manual input event 305 to the operating

system 204. In response to the simulated manual input event 305, the operating system 204 provides a manual input event message 203 to the application 206. In response to the manual input event message 203, the application 206 initiates performance of an appropriate task, such as
 5 implementing a scan function on a radio located within the motor vehicle.

Thus, in the above described system, the application 206, instead of replicating the code corresponding to the manual input, simply generates a simulated manual input event 305 in response to the voice message 303. In this manner, the application 206 is not required to duplicate the code
 10 for responding to both a manual input event message 203 and a voice message 303. This is advantageous in that code replication within the application is avoided. Thus, the application uses the same code to initiate a response to a voice message 303 and a manual input (i.e., hardware) event message 203. In a preferred embodiment, the application generates simulated manual input
 15 events 305 through the use of a system call function (e.g., `keybd_event`) to simulate a particular manual input, e.g., a key press, a key hold and/or a key release. Exemplary voice recognition application C++ code for generating a simulated manual input event 305, according to the present invention, is set forth below:

```

20 case WM_SPCH_RECOG : {
    // NKDbgPrintFW(TEXT("RADIO VOICE REC HOOKS:
    WM_SPCH_RECOG\n"));
    switch (lParam) {
25 case IDVMI_RADIO_TUNEUP: {
        // NKDbgPrintFW(TEXT("RADIO VOICE REC HOOKS:
        Tune right\n"));
        // Key event generated from within app.
        keybd_event(VK_RIGHT,1,0,0);
30 return S_OK;
    }

    case IDVMI_RADIO_TUNEDOWN: {
        // NKDbgPrintFW(TEXT("RADIO VOICE REC HOOKS:
35 Tune left\n"));
        keybd_event(VK_LEFT,1,0,0);
  
```

```

        return S_OK;
    }

    case IDVMI_RADIO_SEEKUP: {
5      // NKDbgPrintfW(TEXT("RADIO VOICE REC HOOKS:
      Seek up\n "));
        keybd_event(VK_UP,1,KEYEVENTF_KEYUP,0);
        return S_OK;
    }

10
    case IDVMI_RADIO_SEEKDOWN: {
        // NKDbgPrintfW(TEXT("RADIO VOICE REC HOOKS:
        Seek down\n "));
        keybd_event(VK_DOWN,1,KEYEVENTF_KEYUP,0);
15        return S_OK;
    }

    // Multiple key events can also be generated
    // in a predetermined sequence.
20    case IDVMI_RADIO_SCANUP : {
        keybd_event(VK_UP,1,0,0);
        keybd_event(VK_UP,1,0,0);
        keybd_event(VK_UP,1,0,0);
        keybd_event(VK_UP,1,0,0);
25        keybd_event(VK_UP,1,0,0);
        keybd_event(VK_UP,1,KEYEVENTF_KEYUP,0);
        // NKDbgPrintfW(TEXT("RADIO VOICE REC HOOKS
        : SCAN Up\n "));
        return S_OK;
30    }

    case IDVMI_RADIO_SCANDOWN: {
        keybd_event(VK_DOWN,1,0,0);
        keybd_event(VK_DOWN,1,0,0);
35        keybd_event(VK_DOWN,1,0,0);
        keybd_event(VK_DOWN,1,0,0);
        keybd_event(VK_DOWN,1,0,0);
        keybd_event(VK_DOWN,1,KEYEVENTF_KEYUP,0);
        // NKDbgPrintfW(TEXT("RADIO VOICE REC HOOKS
40        : SCAN down\n "));
        return S_OK;
    }

    case IDVMI_RADIO_PRESET1: {
45        keybd_event(VK_APCSOFTKEY1,1,0,0);

```

```

        keybd_event(VK_APCSOFTKEY1,1,KEYEVENTF_K
EYUP,0);
//          NKDbgPrintfW(TEXT("AUDIO VOICE REC
HOOKS : TREBLE Up\n "));
5         return S_OK;
    }

case IDVMI_RADIO_PRESET2: {
        keybd_event(VK_APCSOFTKEY2,1,0,0);
10        keybd_event(VK_APCSOFTKEY2,1,KEYEVENTF_K
EYUP,0);
//          NKDbgPrintfW(TEXT("AUDIO VOICE REC
HOOKS : TREBLE Up\n "));
        return S_OK;
15    }

case IDVMI_RADIO_PRESET3: {
        keybd_event(VK_APCSOFTKEY3,1,0,0);
        keybd_event(VK_APCSOFTKEY3,1,KEYEVENTF_K
20        EYUP,0);
//          NKDbgPrintfW(TEXT("AUDIO VOICE REC
HOOKS : TREBLE Up\n "));
        return S_OK;
    }
25

case IDVMI_RADIO_PRESET4: {
        keybd_event(VK_APCSOFTKEY4,1,0,0);
        keybd_event(VK_APCSOFTKEY4,1,KEYEVENTF_K
EYUP,0);
30        //          NKDbgPrintfW(TEXT("AUDIO VOICE REC
HOOKS : TREBLE Up\n "));
        return S_OK;
    }

35    case IDVMI_RADIO_PRESET5: {
        keybd_event(VK_APCSOFTKEY5,1,0,0);
        keybd_event(VK_APCSOFTKEY5,1,KEYEVENTF_K
EYUP,0);
//          NKDbgPrintfW(TEXT("AUDIO VOICE REC HOOKS
40        : TREBLE Up\n "));
        return S_OK;
    }

45    case IDVMI_ONE:
    case IDVMI_TWO:
    case IDVMI_THREE:

```

```

case IDVMI_FOUR:
case IDVMI_FIVE:
case IDVMI_SIX:
case IDVMI_SEVEN:
5 case IDVMI_EIGHT:
case IDVMI_NINE:
case IDVMI_ZERO:
{
    // Select the spoken preset number
10 dwaDirect[index] = GetNumber(IParam);
    if (dwaDirect[index] == -1)
    {
        return S_FALSE;
    }
15
    // NKDbgPrintfW(TEXT("Stored Voice
Number %d in index %d\n "),dwaDirect[index],index);
    index++;
    if ( index >=6)
20 {
        // NKDbgPrintfW(TEXT("Index Maxed out
:: %d\n "),index);
        index=0; pointindex=0;
    }
25
    // check if array filled
    if (pointindex!=0)
    {
        // yes tune
30 DWORD dwFreq = DtrmnFreq();
        // NKDbgPrintfW(TEXT("Voice
Frequency = %d\n "),dwFreq);
        if((dwFreq <= 87700 ) || (dwFreq
>= 107900))
35 {
            WCHAR wch[45];
            wsprintf(wch, L"Invalid Frequency, Try
Again!!");
            BSTR bstr = SysAllocString(wch);
40 m_Speech->Speak(bstr, 0);
            SysFreeString(bstr);
            // NKDbgPrintfW(TEXT("Invalid
frequency try again %d\n "));
            }
45 else
            {

```

```

//                                NKDbgPrintFW(TEXT("Tuning to %d\n
"),dwFreq);
                                m_pRadio->Tune(dwFreq);
                                }
5                                index=0; pointindex=0;
                                }
                                return S_OK;
                                }

10                                case IDVMI_VPOINT:
                                {
                                //    NKDbgPrintFW(TEXT("POINT stored %d\n "),index);
                                pointindex = index;
                                index ++;
15                                }

                                default :
                                if ( index > 5 )
                                {
20                                //    NKDbgPrintFW(TEXT("Index Maxed out ::
                                %d\n "),index);
                                index=0; pointindex=0;
                                }
                                break;
25                                }
                                }

```

Turning to Fig. 4, a flowchart of an information transfer routine 400 is depicted. In step 402 the routine 400 is initiated at which point control transfers to decision step 404. In decision step 404, the processor 102, executing an operating system 204, determines whether a voice input is detected. If so, control transfers from step 404 to step 406 where a voice message 303 is passed from the operating system 204 to a voice recognition application 206. Next, in step 408, the application 206 provides a simulated manual input event 305 to the operating system 204. Then, in step 410, the operating system 204 passes a manual input event message 203 to the application 206 at which point control transfers to step 404.

In step 404, when the processor 102 does not detect a voice input, control transfers to decision step 412. In step 412, the processor 102 determines whether a manual input is detected. If so, control transfers from

step 412 to step 410 where the operating system 204 passes a manual input event message 203 to the application 206. In step 412, when a manual input is not detected, control transfers to step 404. Thus, a routine has been described which illustrates the flow of information between an operating
 5 system and a voice recognition application, according to an embodiment of the present invention.

Figs. 5A-5B depict exemplary block diagrams illustrating the execution of calls to perform a function associated with a voice event according to an embodiment of the present invention and according to the prior art, respectively. As shown in Fig. 5A, in response to a simulated
 10 manual input event (provided from a voice application 502 to an operating system), a manual input event message is provided by the operating system to the application 502. In response to the manual input event message, the application 502 initiates the performance of an appropriate task. For example,
 15 when a user has chosen to adjust the bass of a radio, the application 502 calls a set bass routine 504. The application 502 also calls a slider routine 506, which performs a number of tasks associated with updating a display of the radio. The slider routine 506 increments a slider value, paints the screen with a new slider value and checks if the new slider value is at a maximum or
 20 minimum slider value and, if so, does not update the display.

According to the prior art, as shown in Fig. 5B, a voice application 512 receives a voice message from the operating system. In response to the voice message, the application 512 calls a routine, for example, a set bass routine 514. To update an associated display, the
 25 application 512 must also call a first routine 516, which increments a slider value. Upon returning from the call of routine 516, the application 512 calls a second routine 518, which paints the screen with a new slider value. Upon returning from the call of routine 518, the application 512 then calls a third routine 520 to determine if the new slider value is at a maximum or minimum
 30 slider value. If the maximum or minimum slider value has been reached, then the routine 520 does not update the display before returning to the application

512. It should be appreciated that in implementing such an audio function, according to the prior art, a voice message received by the application is handled in multiple steps. However, according to the present invention, when a voice event is initiated, the voice application 502 provides a simulated
5 manual input event to the operating system. The operating system then responds by providing a manual input event message to the application 502. The application 502 then updates the display by calling a slider routine 506, which is a building block that handles the call independently without requiring the application 502 to further track the process. In contrast, as shown in Fig.
10 5B, the application 512 is required to track each step of the display update process when a voice message is received.

Accordingly, a voice responsive system has been described that initiates control of a device in response to both a manual input and a voice input. The voice responsive system includes a voice recognition application
15 that can initiate a task corresponding to a manual input or a corresponding voice input using the same code. This advantageously reduces the amount of code within the application. The voice responsive system, when implemented within a motor vehicle, advantageously allows an occupant of the vehicle to control various motor vehicle accessories, using both manual input and voice
20 input. Further, reducing the amount of code decreases memory requirements and, as such, reduces the cost of implementing the voice recognition application within an automotive environment (e.g., within a motor vehicle).

The above description is considered that of the preferred embodiments only. Modification of the invention will occur to those skilled
25 in the art and to those who make or use the invention. Therefore, it is understood that the embodiments shown in the drawings and described above are merely for illustrative purposes and not intended to limit the scope of the invention, which is defined by the following claims as interpreted according to the principles of patent law, including the Doctrine of Equivalents.